

EMPLOYING ACTIVE CONTOURS AND ARTIFICIAL NEURAL NETWORKS IN REPRESENTING ULTRASONIC RANGE DATA

Kerem Altun and Billur Barshan

Department of Electrical and Electronics Engineering, Bilkent University
Bilkent, 06800, Ankara, Turkey
phone: + (90-312) 290-2384, fax: + (90-312) 266-4192
email: {kaltun, billur}@ee.bilkent.edu.tr

ABSTRACT

Active snake contours and Kohonen's self-organizing feature maps (SOM) are considered for efficient representation and evaluation of the maps of an environment obtained with different ultrasonic arc map (UAM) processing techniques. The mapping results are compared with a reference map acquired with a very accurate laser system. Both approaches are convenient ways of representing and comparing the map points obtained with different techniques among themselves, as well as with an absolute reference. Snake curve fitting results in more accurate maps than SOM since it is more robust to outliers. The two methods are sufficiently general that they can be applied to discrete point maps acquired with other mapping techniques and other sensing modalities as well.

1. INTRODUCTION

Ultrasonic sensing is widely used in many range finding applications, such as bathymetry and mapping in robotics. Many mapping systems employ ultrasonic sensors due to their low cost as opposed to more expensive and complicated laser and camera systems. Despite their accurate range finding capabilities, the angular resolution of these sensors is poor because of their large beamwidth. Furthermore, multiple and higher-order reflections and crosstalk are known problems associated with ultrasonic transducers, which may cause erroneous readings in the raw ultrasonic data. Thus, signal processing techniques and algorithms must be employed in order to represent the sensor data in an intelligent manner and eliminate these weaknesses inherent to ultrasonic sensors.

In earlier work, there have been two main approaches to the representation of ultrasonic data: feature-based and grid-based. Grid-based approaches do not attempt to make difficult geometric decisions early in the interpretation process unlike feature-based approaches that extract the geometry of the sensor data as the first step. As a first attempt to feature-based mapping, several researchers have fitted line segments to ultrasonic data as features that crudely approximate the room geometry [1, 2, 3]. This approach proved to be difficult and brittle because straight lines fitted to time-of-flight (TOF) data do not necessarily match or align with the world model, and may yield many erroneous line segments. Improving the algorithms for detecting line segments and including heuristics does not really solve the problem. A more physically meaningful representation is the use of *regions of constant depth* (RCDs) as features. RCDs are circular arcs which are natural features of the raw ultrasonic TOF data from specularly reflecting surfaces, first reported in [4], and further elaborated on in [5]. They are obtained by placing a small mark along the line-of-sight at the range corresponding to the measured TOF value. In specularly reflecting environments, the accumulation of such marks usually produces arc-like features.

As a more general approach which is not limited to specularly reflecting surfaces, the angular uncertainty in the range measurements has been represented by ultrasonic arc maps (UAMs) [6] that preserve more information (see Fig. 1(c) for a sample UAM). Note that the arcs in the UAM are different than the arcs corresponding to RCDs. The UAMs are obtained by drawing arcs spanning the beamwidth of the sensor at the measured range, representing the

angular uncertainty of the object location and indicating that the echo-producing object can lie anywhere on the arc. Thus, when the same transducer transmits and receives, all that is known is that the reflection point lies on a circular arc of radius r . More generally, when one transducer transmits and another receives, it is known that the reflection point lies on the arc of an ellipse whose focal points are the transmitting and receiving elements. The arcs are tangent to the reflecting surface at the actual point(s) of reflection. Construction of UAMs is more general in that they can be generated for environments comprised of both specularly and diffusely reflecting (Lambertian) surfaces. On the other hand, construction of RCDs are limited to specularly reflecting environments only. In earlier work, techniques based on the Hough transform have been applied to detect line and point features from these arcs for both air-borne and underwater ultrasonic data [7].

Previously, several techniques have been proposed to process UAMs (Table 1) that reduce the angular uncertainty in the sensor data and eliminate most of the artifacts resulting from multiple and higher-order reflections and crosstalk. All of these techniques result in more accurate maps than the raw UAM. Descriptions of these techniques can be found in [8, 9], or respective references indicated in Table 1. They will not be reviewed here due to space limitations. For the purposes of this paper, we consider each processed UAM as a collection of (usually a large number of) data points, represented as a black-on-white image. In this paper, active snake contours and Kohonen's self-organizing maps (SOM) will be employed for compact representation and evaluation of the results of processing the UAM by each technique and a comparison with a very accurate laser map (considered as absolute reference) will be provided.

Table 1: Different UAM processing techniques.

1	point marking (PM) [4]
2	triangulation-based fusion (TBF) [10]
3	voting and thresholding (VT) [11]
4	directional maximum (DM) [8]
5	morphological processing (MP) [6]
6	Bayesian update (BU) [12]
7	arc transversal median (ATM-org) [13]
8	modified ATM (ATM-mod) [8]

2. FITTING ACTIVE SNAKE CONTOURS TO UAMS

A snake, or an active contour, first introduced by Kass *et al.* [14], can be described as a continuous deformable closed curve. In the literature, snake curves have been commonly used in image processing for edge detection and segmentation [14, 15, 16]. We define a snake as a parametrized closed curve $\mathbf{v}(s) = (x(s), y(s))$, $s \in [0, 1]$, where $x(s)$ and $y(s)$ are functions representing the x and y coordinates of the snake and s is the normalized arc length parameter of the snake curve. The deformation of the snake is controlled by

internal and external forces. Internal forces impose elasticity and rigidity constraints on the curve, whereas external forces depend on the image and they try to stretch or shrink the curve to fit to the data. The total energy of the snake curve is given by the functional

$$E = \int_0^1 \left[\frac{1}{2} \left(\alpha \left\| \frac{d\mathbf{v}}{ds} \right\|^2 + \beta \left\| \frac{d^2\mathbf{v}}{ds^2} \right\|^2 \right) + U(\mathbf{v}) \right] ds \quad (1)$$

where α is the elasticity parameter and β is the rigidity parameter, taken as constants, and $\|\cdot\|$ denotes the 2-norm. The first derivative term in Eqn. (1) penalizes long curves, whereas the second derivative term penalizes sharp curvatures.

The external energy component is denoted by $U(\mathbf{v})$, where U is a potential function that depends on the image data. In general, the selection of the potential function varies depending on the application. However, it must be minimum on the image edges if the snake is to be used for edge detection, segmentation, or finding the boundaries of an environment. Kass *et al.* suggest using the negative of the image gradient magnitude as a potential function [14]. However, this is only feasible if the snake is initialized close to the image boundaries. Otherwise, the snake curve would be stuck in local minima or in a flat region of the potential function. Filtering the image with a Gaussian low-pass filter is also suggested in [14] to increase the capture range of the snake, but this causes the edges to become blurry, thus reducing the accuracy. Another solution proposed in [16] is using a *distance map* as a potential function to increase the capture range of the contour, which is the approach used in this study. In the literature, using force balance equations rather than the energy-based approach has also been reported [17]. In this work, we chose to use a potential function for the external energy term. To find the curve that minimizes the energy functional in Eqn. (1), calculus of variations can be used. The minimizing curve should satisfy the following Euler-Lagrange equation [14]:

$$\alpha \frac{d^2\mathbf{v}(s)}{ds^2} - \beta \frac{d^4\mathbf{v}(s)}{ds^4} - \nabla U(\mathbf{v}(s)) = 0 \quad (2)$$

Although it may be possible to solve this equation analytically for some special cases, a general analytical solution does not exist. The common practice is to initialize an arbitrary time-dependent snake curve $\mathbf{v}(s, t)$. Eqn. (2) is then set equal to the time derivative of the snake, where a solution will be found when the time derivative vanishes. That is,

$$\alpha \frac{\partial^2\mathbf{v}(s, t)}{\partial s^2} - \beta \frac{\partial^4\mathbf{v}(s, t)}{\partial s^4} - \nabla U(\mathbf{v}(s, t)) = \frac{\partial\mathbf{v}(s, t)}{\partial t} \quad (3)$$

These equations are then discretized to find a numerical solution. The snake is treated as a collection of discrete points joined by straight lines and is initialized on the image. Approximating the derivatives by finite differences, the evolution equations of the snake reduce to the following [15]:

$$\mathbf{x}(n+1) = (\mathbf{A} + \gamma\mathbf{I})^{-1} \left(\gamma\mathbf{x}(n) - \kappa \frac{\partial U}{\partial x} \Big|_{(\mathbf{x}(n), \mathbf{y}(n))} \right) \quad (4)$$

$$\mathbf{y}(n+1) = (\mathbf{A} + \gamma\mathbf{I})^{-1} \left(\gamma\mathbf{y}(n) - \kappa \frac{\partial U}{\partial y} \Big|_{(\mathbf{x}(n), \mathbf{y}(n))} \right) \quad (5)$$

Here, n is the current time (or iteration) step, $\mathbf{x}(n)$ and $\mathbf{y}(n)$ represent the coordinates of the discrete points on the snake at time n , γ is the Euler step size, and κ is the external force weight. \mathbf{I} is the identity matrix of appropriate size and \mathbf{A} is a penta-diagonal banded matrix that depends on α and β . The dimensions of \mathbf{A} and \mathbf{I} are determined by the number of points on the snake, which may change as the algorithm is executed.

3. FITTING SELF-ORGANIZING MAPS TO UAMS

Another method used in map representation and evaluation of the different techniques is the self-organizing map introduced by Kohonen [18]. This neural network utilizes a form of unsupervised learning, and is used in applications where the topology of the data is to be learned. SOMs for curve and surface reconstruction have been used in applications such as CAD modeling of objects having irregular shape. In this paper, we use the SOM for fitting curves to the ultrasonic map points obtained with the different UAM processing techniques.

A SOM is an artificial neural network with two layers. The two neurons at the input layer represent the x and y coordinates of an extracted point on the map, and the output layer is arranged as a 1-D array, representing the points on the curve to be fitted to the map. The two weights associated with each output neuron represent the x and y coordinates of the corresponding point on the fitted curve. For each input map point, the winning neuron is determined as the closest point on the curve to that input. Thus, for the input map point (x, y) , output neuron weights (w_{1i}, w_{2i}) , and a total of N points on the curve, the index c of the winning neuron is given by:

$$c = \arg \min_{i=1, \dots, N} \sqrt{(x - w_{1i})^2 + (y - w_{2i})^2} \quad (6)$$

Through the use of a Gaussian function $g_{0, \sigma(n)}(\cdot)$, updating the weights is done such that the weight update of one neuron also affects the neighboring neurons. Then, for all neurons $i = 1, \dots, N$, the weight update rule is

$$\mathbf{w}_i(n+1) = \mathbf{w}_i(n) + \alpha(n) g_{0, \sigma(n)}(|i - c|) (\mathbf{d}(n) - \mathbf{w}_i(n)) \quad (7)$$

where n is the iteration step, $\mathbf{w}_i(n)$ is the 2×1 weight vector of neuron i , $\mathbf{d}(n)$ is the vector with the x and y coordinates of the input point on the map, $\alpha(n)$ is the time-dependent learning rate, and $g_{0, \sigma(n)}(\cdot)$ is a 1-D Gaussian function with zero mean and standard deviation $\sigma(n)$. The Gaussian has a value of one at its peak point that occurs for $i = c$, and decreases gradually.

4. EXPERIMENTS

To demonstrate our methodology, the mapping results of different UAM processing techniques, listed in Table 1 and described in detail in [8], are used. Each of these techniques results in a different set of map points to which both a snake curve and a SOM are fitted.

To define a potential function suitable for our purposes, we use distance functions and distance transforms in the following sections. Therefore, let us introduce a generic distance function at this stage. Let $P \subset \mathbb{R}^2$ be a finite set of arbitrary points of the plane. We define a distance function $D_P(x, y) : \mathbb{R}^2 \rightarrow \mathbb{R}^{\geq 0}$ for each point (x, y) on the plane as the minimum of the Euclidean distances of that point to all the points in the set P :

$$D_P(x, y) = \min_{p \in P} \sqrt{(x - p_x)^2 + (y - p_y)^2} \quad (8)$$

where (p_x, p_y) are the coordinates of a point in the set P . The Euclidean distance transform is computationally costly, and a number of algorithms and other distance transforms have been proposed in the literature to approximate it [20]. In this study, the Euclidean distance transform is implemented in its original form.

4.1 Results of snake fitting

The potential function used in this study is based on the Euclidean distance transform, as suggested in [16]. As stated before, we represent the processed UAMs and the reference laser data as black pixels on white background. Euclidean distance map is defined for all points on the image as the Euclidean distance to the nearest black pixel. Let each set of processed UAM data points be denoted as M_k ,

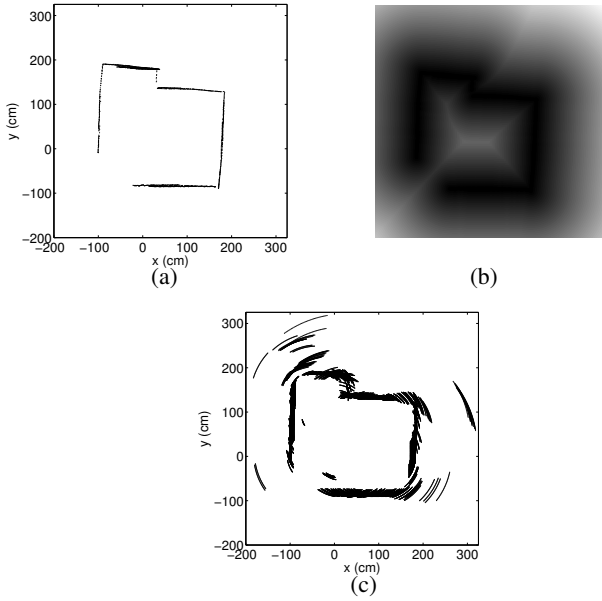


Figure 1: (a) Original laser map, (b) distance map with respect to the laser data, (c) and the raw UAM.

where k corresponds to one of the UAM processing techniques indexed in Table 1. For compatibility, let the set of laser data points be denoted as M_0 . Then, for the laser map ($k = 0$) and for the k^{th} ultrasonic map ($k = 1, \dots, 8$), the potential function used in the fitting of the k^{th} snake is selected as

$$U_k(x, y) = D_{M_k}(x, y) \quad k = 0, 1, \dots, 8 \quad (9)$$

for all points (x, y) on the image. Note that the value of the potential function is zero for those points on the image corresponding to the extracted map and increases gradually with increasing distance of the point (x, y) from the map points.

An example map of a room acquired with a structured-light laser system is shown in Fig. 1(a). This is the original laser data which is quite accurate, and is used as the absolute reference to compare the methods given in Table 1. The corresponding Euclidean distance map is shown in Fig. 1(b), which is drawn by rescaling the values of the potential function to be between 0 and 255. In the distance map, the darkest points of value 0 correspond to a distance of 0 and, after normalization, the lightest points correspond to the value 255.

The values for the parameters in Eqns. (4) and (5) are selected as $\alpha = \gamma = 1$, $\beta = 0.1$ and $\kappa = 2.5$. Selecting $\beta = 0.1$ enforces the second derivative in the energy term to have less weight, thus allowing sharp corners in the snake. The snake curves fitted to the laser data and the processed UAMs are illustrated in Fig. 2. In the different parts of the figure, the black features correspond to map points obtained with a particular UAM processing technique. The blue curves are the snakes fitted to these map points. The red curve is the snake fitted to the laser data, which is the same in each part of the figure and is included as a reference for visual comparison. As a note, there was an opening in the lower-left corner of the room from which no ultrasonic data were collected. Therefore, the part of the snake curve in that region is not drawn in the figure and is not included in the error calculations.

The snake is initialized as a circle whose center is at (30, 55) having a radius of 185 units so that it encompasses the room boundary. Then, the snake is evolved for 250 iterations. After each iteration, the points on the snake are checked for uniformity. The distance between any two neighboring points is maintained between 2–4 units, determined experimentally. That is, after each iteration, the points are deleted or created as required by this constraint. We allow the snake to converge to outlier points caused by multiple and higher-order reflections or crosstalk to provide a fair evaluation of the different techniques.

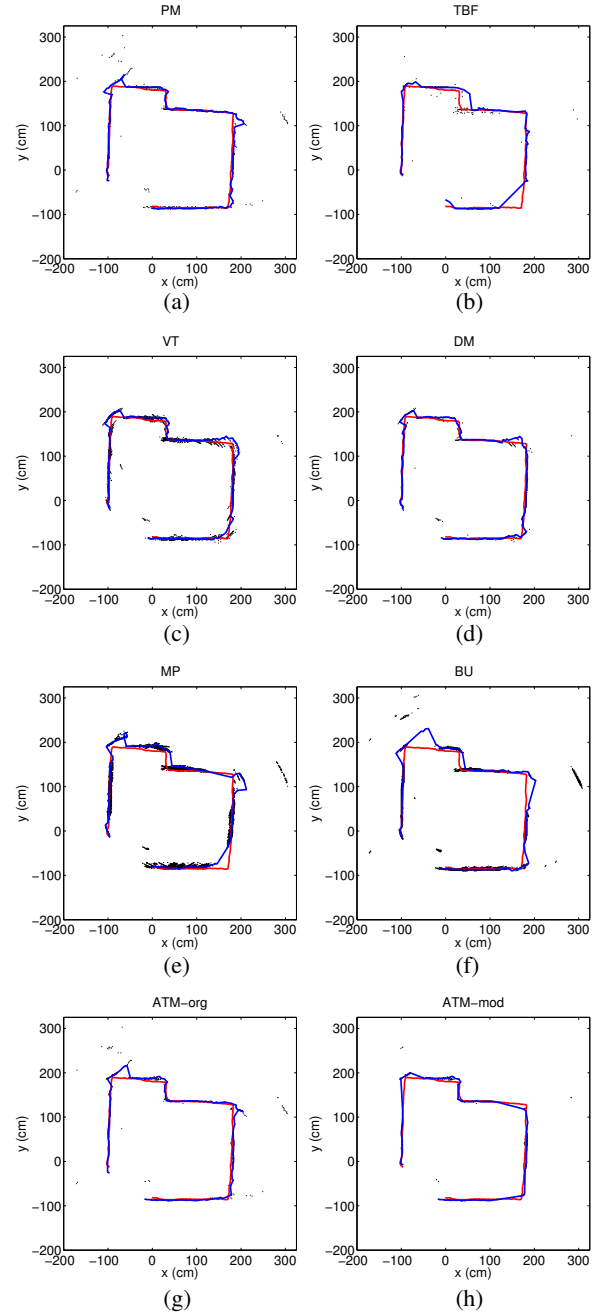


Figure 2: Results of snake fitting for (a) PM, (b) TBF, (c) VT, (d) DM, (e) MP, (f) BU, (g) ATM-org, and (h) ATM-mod.

We define the error of the fit as the average value of the snake points on the corresponding distance map, i.e.,

$$\mathcal{E}(n) = \frac{1}{N_k(n)} \sum_{j=1}^{N_k(n)} D_{M_k}(x_{jk}(n), y_{jk}(n)) \quad k = 0, \dots, 8 \quad (10)$$

where n is the iteration step, $(x_{jk}(n), y_{jk}(n))$ are the coordinates of point j on snake k at iteration n , and $N_k(n)$ is the number of points on snake k at step n . We calculate and store this error at each iteration. Then, the snake curve which results in the minimum error is determined and selected as the snake that best represents the corresponding map points.

In this paper, we used a fixed number of iterations (250) and selected the snake that gives the minimum error. Our observations reveal that the error decreases to a certain value between iterations

100 and 150 and then oscillates around that value, which does not affect the results significantly. In a practical application, it is also possible to take the error at the end of a fixed number of iterations or to set an error threshold to stop the iterations when the error goes below the threshold. The error values at the end of 100 iterations are reported in [21].

Let the set of points on the snake curve fitted to the map points obtained with the k^{th} UAM processing technique be denoted as S_k . That is, the k^{th} snake is represented as a collection of points (x_{jk}, y_{jk}) , $j = 1, \dots, N_k$, where N_k is the total number of points on snake S_k . The set of points of the snake fitted to the original laser data is denoted as S_0 .

We define an error measure that determines the closeness between the processed UAM snake S_k and the laser snake S_0 . It is calculated by finding the distance of every point on the snake S_k to the nearest point on the snake S_0 and averaging these distances. Using the notation in the beginning of this section, the minimum distance of point j on snake S_k to the laser snake is given by $D_{S_0}(x_{jk}, y_{jk})$. Then, the error is given as:

$$\mathcal{E}_k = \frac{1}{N_k} \sum_{j=1}^{N_k} D_{S_0}(x_{jk}, y_{jk}) \quad k = 0, \dots, 8 \quad (11)$$

Note that $\mathcal{E}_0 = 0$ by definition. The errors for the different methods are tabulated in Table 2. According to the results, ATM-mod and DM techniques have the smallest errors, and MP and BU perform the worst. The remaining techniques are comparable to the PM method.

Table 2: Error values for snake curve fitting and SOM.

method	snake error	SOM error
PM	4.11	19.23
TBF	4.18	8.72
VT	3.77	6.55
DM	3.22	4.29
MP	7.00	20.96
BU	6.23	29.35
ATM-org	4.26	20.88
ATM-mod	3.20	6.89

The snake curves in this study are initialized outside the boundaries of the room. Initializing the snake within the boundaries of the room is also a possibility. In this case, the spurious points outside the boundaries would not affect the snake curve as much, allowing it to follow the boundaries of the room more closely. However, this would not result in a fair comparison between the techniques in terms of the amount of spurious points left after UAM processing. Because of the first derivative term in the energy expression, the snake curve tends to shrink rather than expand. This fact should also be taken into account in determining the initial location of the snake curve. In addition, in some mapping applications, one may not be that free in choosing the initial location. For example, in a room with many obstacles close to the room boundaries, it would be essential to initialize the snake curve outside in order to represent the boundaries of the room correctly. However, if the detection of the obstacles is of more importance, one could choose to initialize the snake inside the boundaries. In fact, some applications may require both. The choice for the initial location should depend on the configuration of obstacles and the free space.

4.2 Results of fitting self-organizing maps

We initialize a SOM with 160 neurons as a circle outside the boundaries of the room, with center at (30, 55), and having a radius of 185 pixels. The center and the radius are chosen the same as in the snake

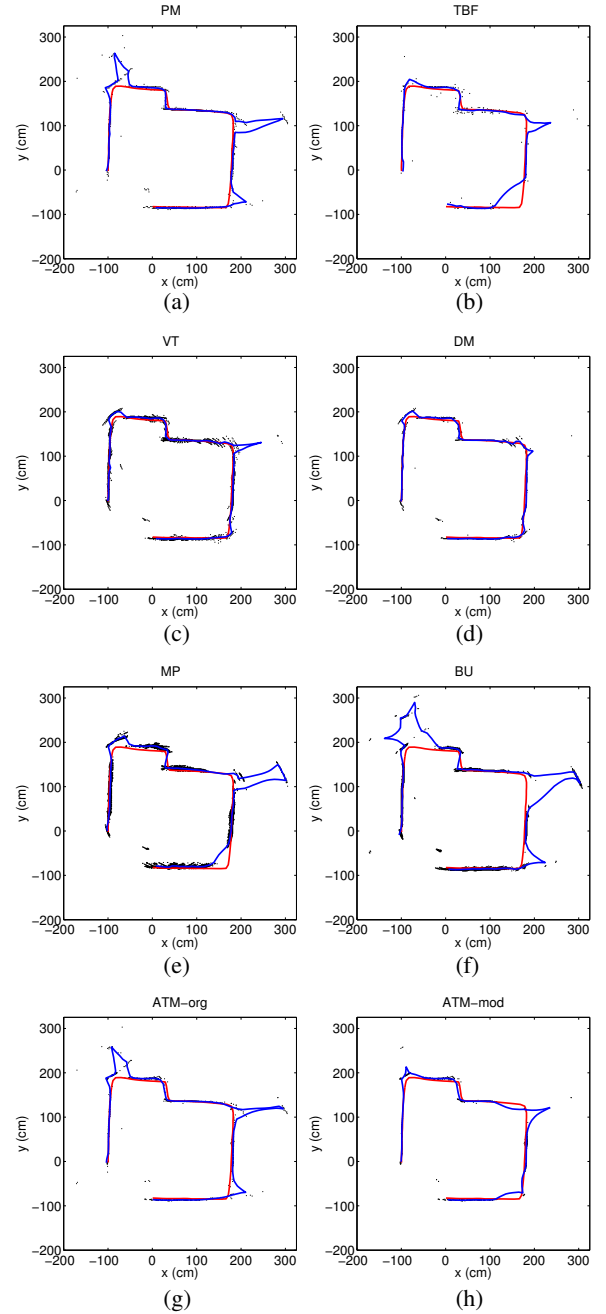


Figure 3: Results of fitting SOMs for (a) PM, (b) TBF, (c) VT, (d) DM, (e) MP, (f) BU, (g) ATM-org, and (h) ATM-mod.

curve fitting procedure. The learning rate is initialized as 0.5, and the Gaussian is initialized with zero mean and standard deviation 8.

The learning rate and the standard deviation are adaptive; we reduce the learning rate to 90% at each epoch and the standard deviation is reduced to about 80% every two epochs. The network converges in 25 epochs. Similar to the case with the snake curve, the distance between neighboring points on the curve is maintained between 10–16 units. The curves fitted to the map obtained by the ATM-org technique is highly affected by the outlier points, unlike the snake curve fitted for that technique. Curves generated

The error criterion defined in the previous subsection is considered for this case as well, with the same definition, using the curves fitted by SOM method. The results are given in Table 2 where SOM errors are, in general, larger than the snake curve fitting errors. It can also be observed that the SOM curve fitted to the map obtained by the ATM-org technique is highly affected by the outlier points, unlike the snake curve fitted for that technique. Curves generated

by fitting SOM are not constrained by length or curvature like the snake curves, thus, they are more likely to fit to the outlier points. This results in larger error values. Note that these error measures only take into account the statistical error. Other error measures that take topology into account have been studied in the literature [19]. However, in our work, such error measures are not considered. That is because our primary aim in using SOMs in evaluation is to also consider the outlier points in the map, which may not have been taken into account if topological error is used.

5. DISCUSSION

Looking at the error values and observing Figs. 2 and 3, it is seen that VT, DM and ATM-mod methods eliminate most of the artifacts in the ultrasonic data resulting from multiple and higher-order reflections, crosstalk, and erroneous measurements. PM, MP, BU and ATM-org methods cannot eliminate those artifacts as much, resulting in larger errors. This can be observed more clearly in Fig. 3.

The two methods used for map representation differ in many aspects, as can be observed in Figs. 2 and 3. Snake curves try to minimize their internal energy (basically length and curvature) and a potential function that is minimum on the data points simultaneously. SOMs, on the other hand, try to fit a curve considering all data points and they are not constrained by length or curvature. Snake curves are more sensitive to changes in the initialization than SOMs, but SOMs are sensitive to the order in which the data points are input to the neural network. In our implementation of SOM, the points were input randomly. In snake curve fitting, the evolution equations (4) and (5) are used to update the position of the snake.

In our experiments, we also recorded computation times for fitting both curves. In snake fitting, the time to compute the distance map varies approximately linearly with the data size. It takes 0.01 seconds per data point to form the distance map. Once the distance map is formed, the time to fit the snake is roughly constant. The computation time for fitting SOMs varies roughly linearly with the number of data points, being about 0.1 seconds per data point. Thus, for maps with few data points (TBF and ATM-mod), fitting a SOM takes less time than fitting a snake curve. For other maps, fitting a SOM takes longer.

6. CONCLUSION

We have presented two approaches to compactly and efficiently represent the maps obtained by processing the UAMs with different techniques. The representation of the map points with snake curves or SOM makes it possible to compare maps obtained with different techniques among themselves, as well as with an absolute reference. The results of our work suggest that using active snake contours is a better way of representing ultrasonic maps than SOM, due to their relative robustness and insensitivity to the outlier map points. Our purpose in this paper was efficient representation of ultrasonic maps and the performance evaluation of UAM processing techniques through the use of a demonstrative example. The two approaches are capable of fitting accurately to features with high curvature, such as edges and corners, as well as smoother features, such as planar walls, in typical indoor environments. They can be employed to fill the erroneous gaps in discrete point maps. The two methods are sufficiently general that they can be applied to map data points acquired with other mapping techniques and other sensing modalities as well. The results can be extended to 3-D data by fitting 3-D shapes. Another possible extension of this work would be the automatic determination of the appropriate number of curves or shapes to be fitted to a given set of extracted map points using clustering techniques. Determining whether the curves should be open or closed, or the shapes convex or concave, and the initialization of the multiple curve parameters are other challenging issues.

Acknowledgment

This work is supported by TÜBİTAK grant EEEAG-105E065.

REFERENCES

- [1] J. L. Crowley, "Navigation for an intelligent mobile robot," *IEEE Trans. Robot. Automat.*, RA-1(1):31–41, Mar. 1985.
- [2] W. Gex and N. Campbell, "Local free space mapping and path guidance," *Proc. IEEE Int. Conf. Robot. Automat.*, pp. 424–431, Mar. 1987.
- [3] M. Drumheller, "Mobile robot localization using sonar," *IEEE Trans. Pattern Anal. Mach. Intell.*, PAMI-9(2):325–332, Mar. 1987.
- [4] R. Kuc and M. W. Siegel, "Physically-based simulation model for acoustic sensor robot navigation," *IEEE Trans. Pattern Anal. Mach. Intell.*, PAMI-9(6):766–778, Nov. 1987.
- [5] J. J. Leonard and H. F. Durrant-Whyte, *Directed Sonar Sensing for Mobile Robot Navigation*, Boston, MA: Kluwer Academic Publishers, 1992.
- [6] D. Başkent and B. Barshan, "Surface profile determination from multiple sonar data using morphological processing," *Int. J. Robot. Res.*, 18(8):788–808, Aug. 1999.
- [7] J. D. Tardós, J. Neira, P. M. Newman and J. J. Leonard, "Robust mapping and localization in indoor environments using sonar data," *Int. J. Robot. Res.*, 21(4):311–330, Apr. 2002.
- [8] B. Barshan, "Directional processing of ultrasonic arc maps and its comparison with existing techniques," *Int. J. Robot. Res.*, 26(8):797–820, Aug. 2007.
- [9] B. Barshan, "A comparative study on the processing of ultrasonic arc maps," *Proc. EUSIPCO'08*, Aug. 2008.
- [10] O. Wijk and H. I. Christensen, "Triangulation-based fusion of sonar data with application in robot pose tracking," *IEEE Trans. Robot. Automat.*, 16(6):740–752, Dec. 2000.
- [11] B. Barshan, "Ultrasonic surface profile determination by spatial voting," *Electron. Lett.*, 35(25):2232–2234, 9 Dec. 1999.
- [12] A. Elfes, "Sonar based real-world mapping and navigation," *IEEE Trans. Robot. Automat.*, RA-3(3):249–265, Jun. 1987.
- [13] H. Choset, K. Nagatani, and N. Lazar, "The arc-transversal median algorithm: a geometric approach to increasing ultrasonic sensor azimuth accuracy," *IEEE Trans. Robot. Automat.*, 19(3):513–522, Jun. 2003.
- [14] M. Kass, A. Witkin, and D. Tersopoulos, "Snakes: Active contour models," *Int. J. Comput. Vision*, 1(4):321–331, Jan. 1988.
- [15] S. Menet, P. Saint-Marc, and G. Medioni, "Active contour models: Overview, implementation and applications," *Proc. IEEE Int. Conf. Syst. Man Cybern.*, pp. 194–199, Nov. 1990.
- [16] L. D. Cohen and I. Cohen, "Finite element methods for active contour models and balloons for 2-D and 3-D images," *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(11):1131–1147, Nov. 1993.
- [17] C. Xu and J. L. Prince, "Snakes, shapes and gradient vector flow," *IEEE Trans. Image Process.*, 7(3):359–369, Mar. 1998.
- [18] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biol. Cybern.*, 43(1):59–69, Jan. 1982.
- [19] G. S. Kumar, P. K. Kalra and S. G. Dhande, "Curve and surface reconstruction from points: an approach based on self-organizing maps," *Appl. Soft Comput.*, 5(1):55–66, Dec. 2004.
- [20] G. Borgefors, "Distance transformations in digital images," *CVGIP*, 34(3):344–371, Jun. 1986.
- [21] K. Altun and B. Barshan, "Performance evaluation of ultrasonic arc map processing techniques by active snake contours," in *Springer Tracts in Advanced Robotics*, H. Bruyninckx, L. Preucil, M. Kulich (eds.), European Robot. Symp. 2008, 44:185–194, Springer, Berlin, Heidelberg, Feb. 2008.